Statistical and Graph Theoretical Approaches to Semantic Tagging of Unstructured Text for BKC

Nagiza F. Samatova
Computer Science and Mathematics Division
Oak Ridge National Laboratory
samatovan@ornl.gov

DHS PI Meeting February 16, 2005

Acknowledgements

- This work is funded by the Department of Homeland & Security

- Tom Slezak, Terence Critchlow, David Butler, and the rest of the LLNL team for very productive collaboration

- The six anonymous users who participated in manual performance evaluation tests

- ORNL team for very hard and creative work on this project:
    - Praveen Chandramohan
    - Ramya Krishnamurthy
    - Rajesh Munavalli
    - Hoony Park
    - Chris Symons

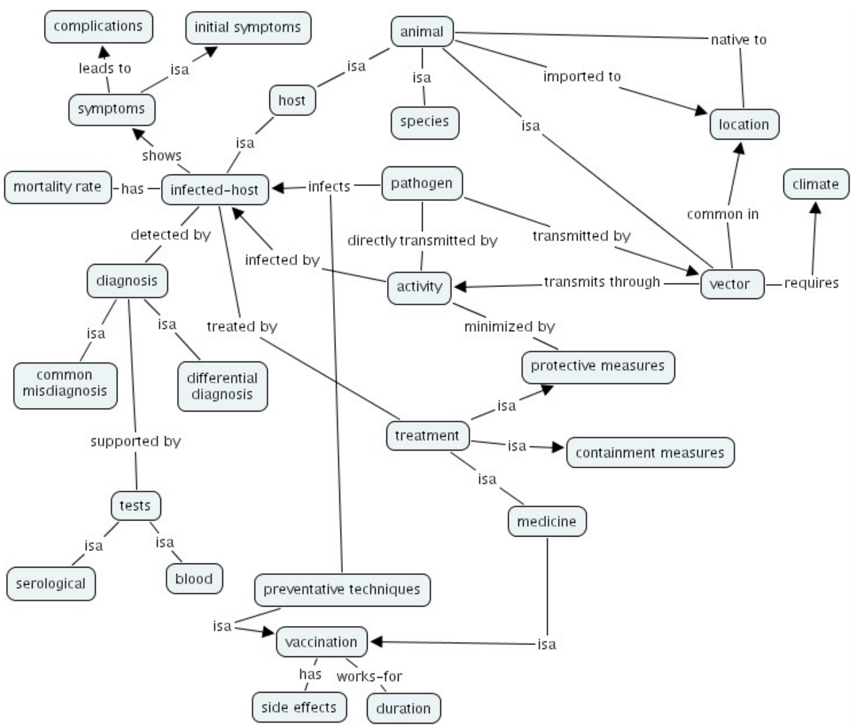
Information Extraction & Semantic Tagging

"Some information, such as endemic countries/locales, etc. is included, but in the text areas… It looks like there could be good additional information gotten from this site" **- excerpt from Susan Hazlett's email**

"As Susan notes, a lot of good stuff is buried in text....“ **- excerpt from Tom Slezak's email**

- Over **100 data sources** were identified to be part of the BKC. Most of them contain rich information in **free text**
- **Manual** reading and curation of textual information is **a challenge**!
- Documents tend to have information that maps to multiple concepts across multiple domains (**ambiguity challenge**)
- Extracting information, mapping them to concepts, and deriving relations between them is **a daunting task**!

A Schematic of the BKC Semantic Graph



Our Goal

To enrich the BKC with information from free text in a "query-friendly" format.

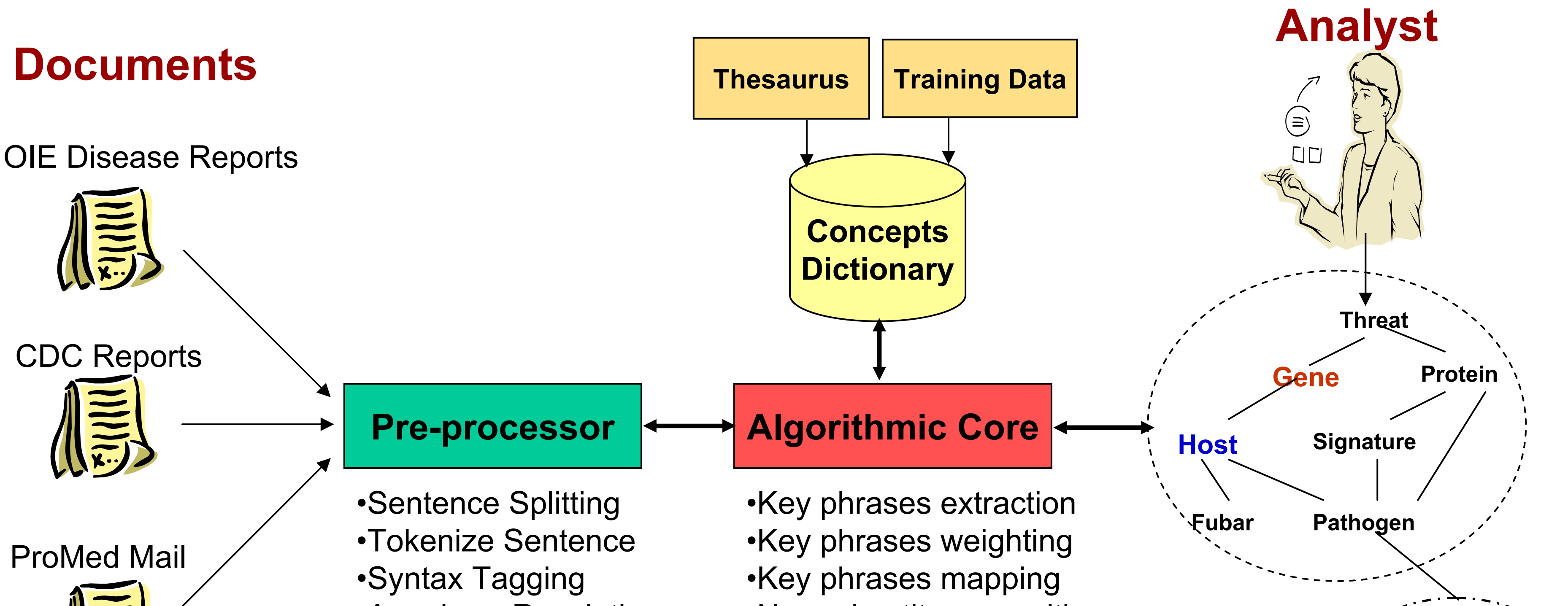
By providing advanced capabilities:

- To extract information relevant to BKC.
- To map the extracted information into respective concepts in the semantic graph.
- To enhance knowledge with Named Entity Recognition for entities critical to DHS.
- To facilitate efficient query over the semantic graph.

Utilizing ORNL expertise in:

- Text Analysis
- Scalable Data Analysis Algorithms
- Parallel Graph Matching Algorithms

System Overview



- Sentence Splitting
- Tokenize Sentence
- Syntax Tagging
- Anaphore Resolution
- Stop words removing
- Stemming
- N-gram generation

- Key phrases extraction
- Key phrases weighting
- Key phrases mapping
- Named entity recognition
- Efficient graph algorithms
- Novel concepts discovery
- Relationships extraction

Foot and Mouth Disease
A virus of the family **Picornaviridae, genus *Aphthovirus***. Seven immunologically distinct serotypes: A, O, C, SAT1, SAT2, SAT3, Asia1.
Hosts: **Bovidae (cattle, zebus, domestic buffaloes, yaks), sheep, goats**, swine, all wild ruminants and suidae. **Camelidae (camels, dromedaries, llamas, vicunas**) have low susceptibility. FMD is endemic in parts of **Asia, Africa, the Middle East and South America** (sporadic outbreaks in free areas)

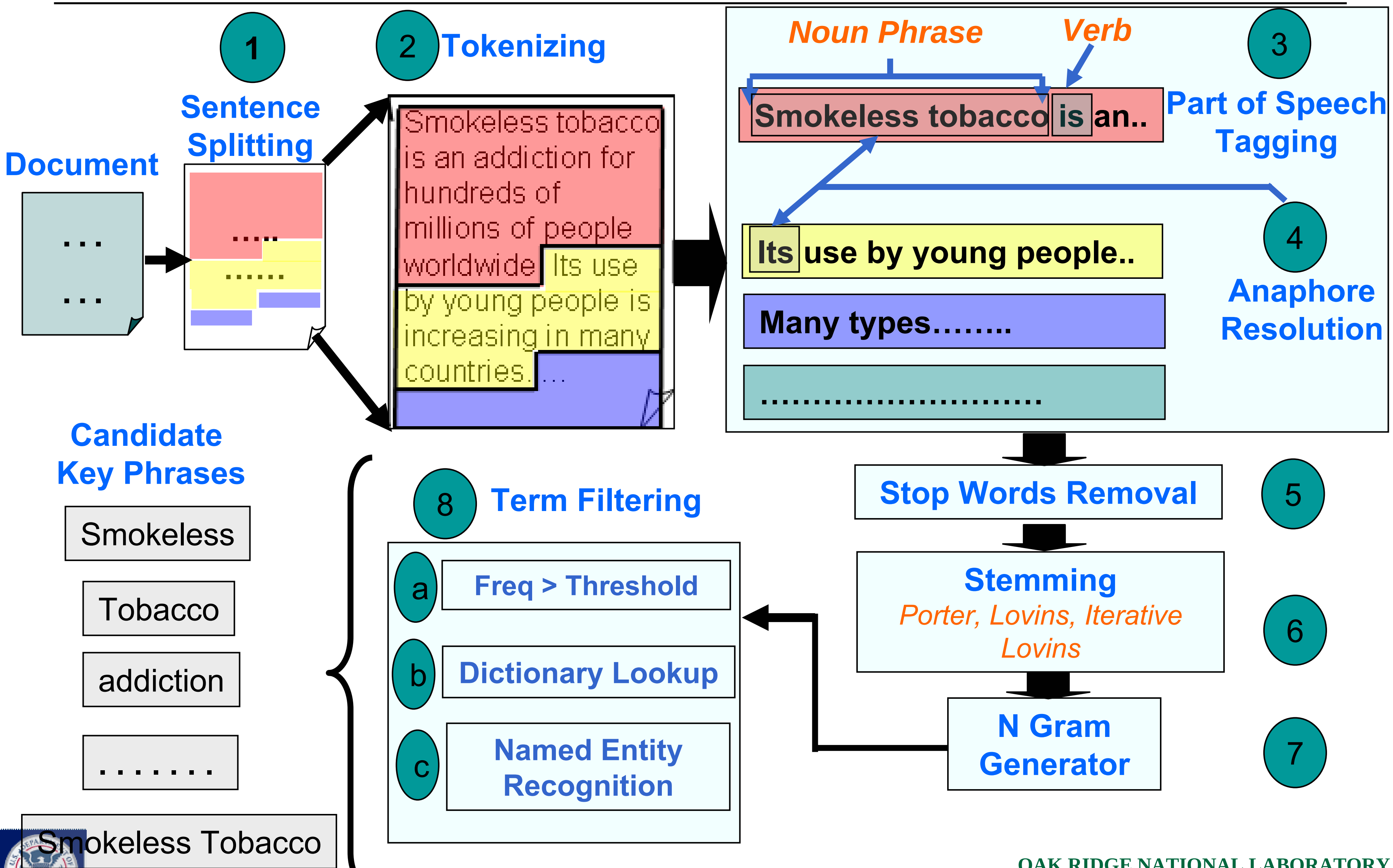
Intelligent Preprocessing is Critical

- Natural Language is usually complex and often ambiguous.
- Many common writing tendencies can confuse automatic methods, and contextual clues utilized by humans are often extremely difficult for a computer to recognize.
- Therefore, intelligent preprocessing methods are crucial to text-analysis applications.
- Important preprocessing stages in our framework include the following:
  - Identifying Coherent Phrases
  - Dealing with Synonymous Phrases
  - Word-Sense Disambiguation
  - Clustering of Related Terms
- Preprocessing can improve the performance of text analysis algorithms by 15-20%

Jack and Jill went up the hill. She stayed up, but **he** fell back down.

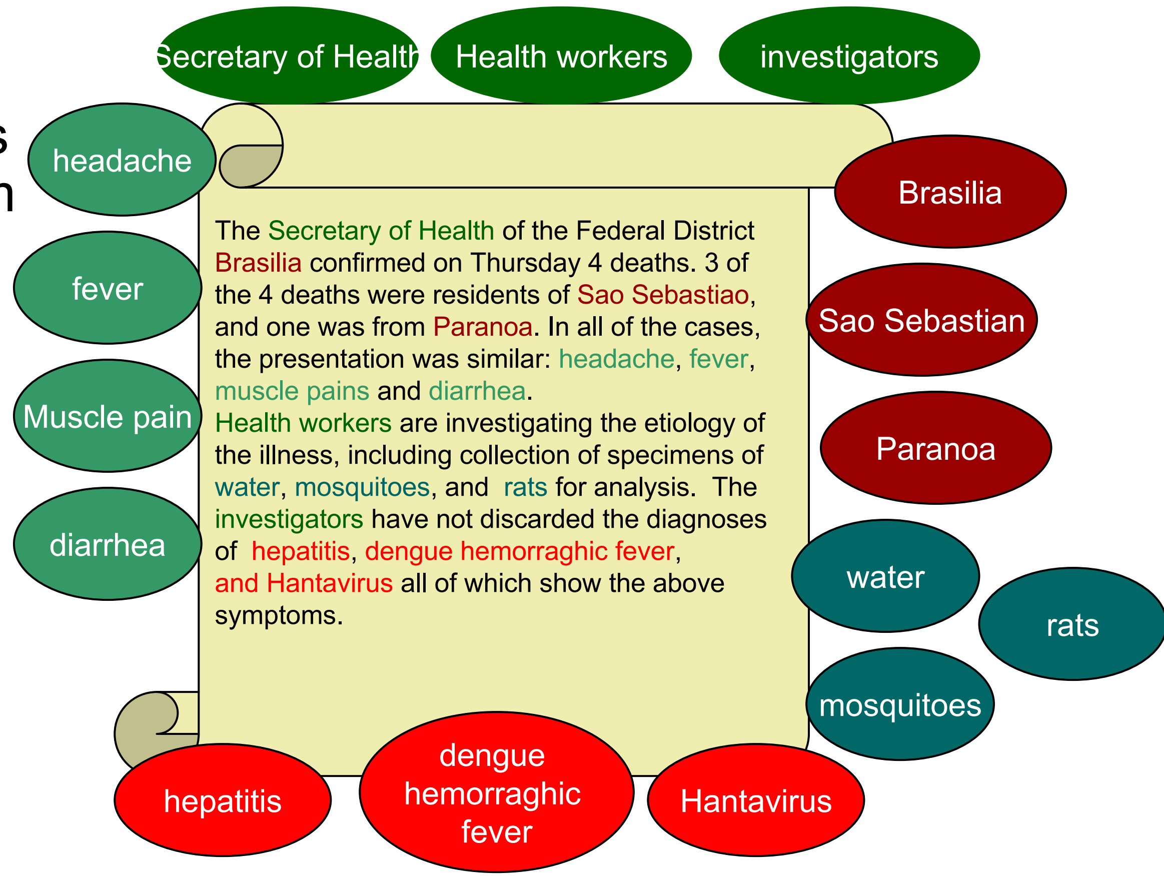
Jack and Jill went up the hill. Jill stayed up, but **Jack** fell back down.

ORNL Preprocessing Package within BKC



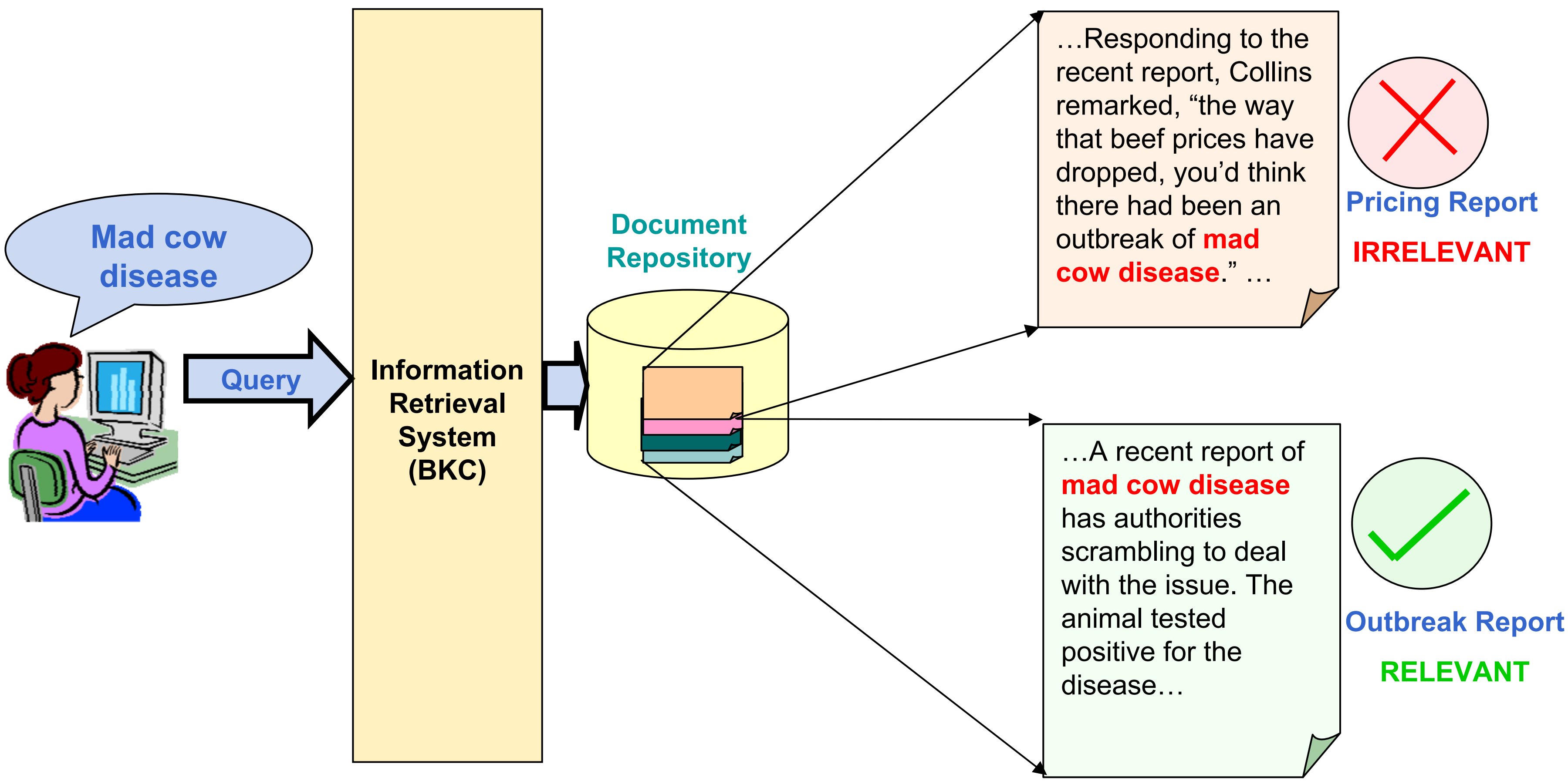
Key Phrases Extraction and Weighting

- Key phrases extraction is often the first step towards extracting information from free text documents.
- Key phrases provide a reasonable understanding of the document content.
- Appropriate weights give the relevance of a document to a particular topic.



They Facilitate Documents Query & Retrieval

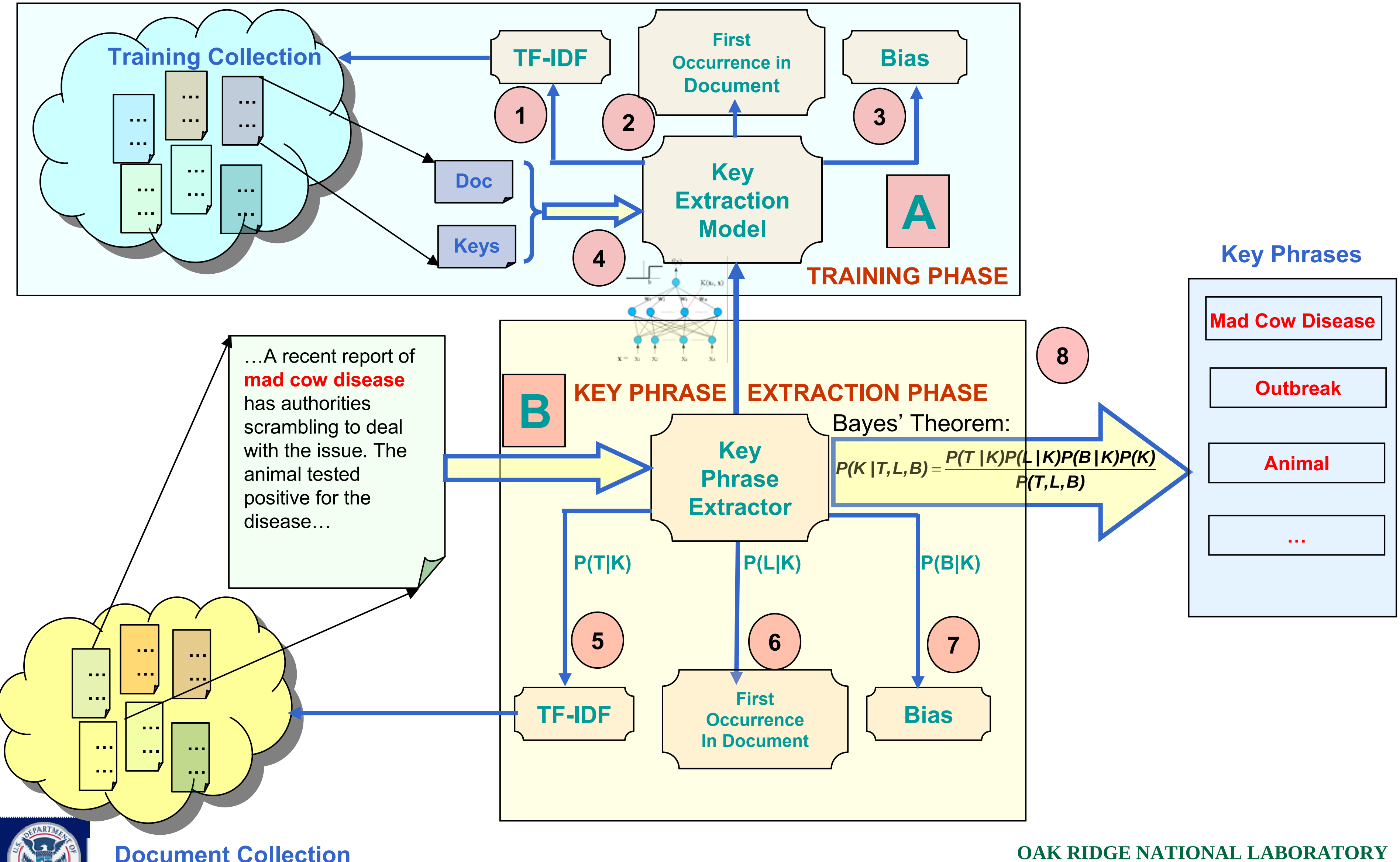
- An important goal is to find relevant documents while avoiding irrelevant documents
- It is not sufficient to simply search for the presence of desired terms.



Approaches to Key Phrases Extraction – Corpus-Dependent and Corpus-Independent Methods

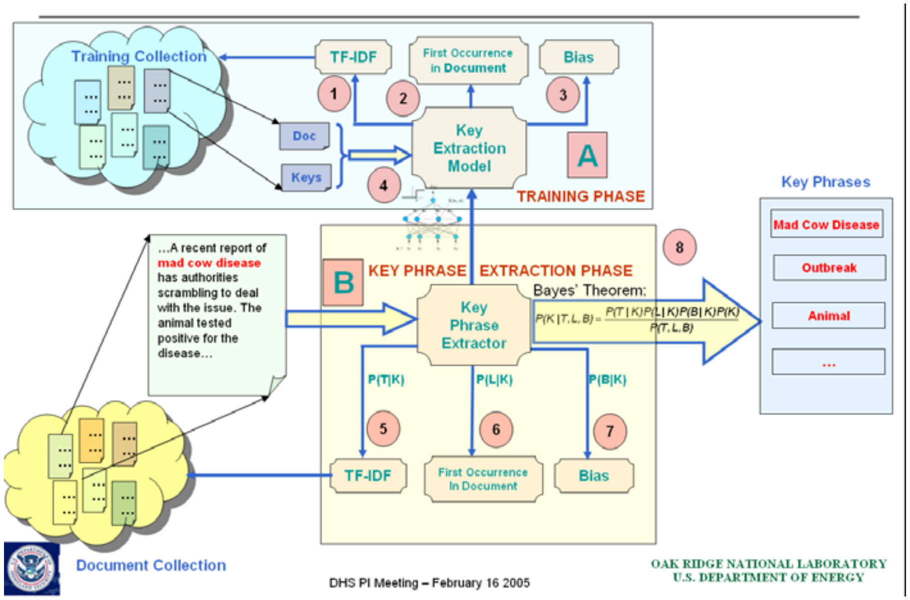
- Each has its own advantages.
- A **corpus dependent** approach can be very useful when documents come from the same source and usually pertain to related topics.
  - We developed a Naïve Bayesian classifier method for situations that allow a corpus-dependent approach.
- A **corpus independent** approach can be very useful if the source of the document is not very consistent and the document could belong to a variety of domains.
  - We developed a term co-occurrence based algorithm for situations that call for a single-document method.

ORNL Corpus-Dependent Key Phrase Extraction



Salient Features – Corpus Dependent Algorithm

- Utilizes domain-specific dictionaries relevant to BKC as a basis for the bias in the Corpus Dependent Method.
- Provides marked improvement in the observed keyphrase extraction.
- Allows identification of documents relevant to BKC without forcing inclusion of documents simply because they contain a related term.



ORNL Corpus-Independent Key Phrase Extraction

$$\chi'^2(w) = \sum_{c\in G}\left\{\frac{(freq(w,c)-n_w p_c)^2}{n_w p_c}\right\} - \max_{c\in G}\left\{\Big(\frac{(freq(w,c)-n_w p_c)^2}{n_w p_c}\right\}$$

Less frequent but important words undetected by TF method

Smokeless tobacco is an addiction for hundreds of millions of people worldwide. Use by young people is increasing in many countries. Many types of smokeless tobacco are marketed for oral or nasal use. All contain nicotine and nitrosamines. DNA and haemoglobin adducts are commonly detected in tobacco users

Tobacco users are exposed to differing levels of nitrosamines. These are formed mainly by nitrosation of nicotine and other tobacco alkaloids during the curing and processing of tobacco, and additional amounts are formed during smoking……

1 Top 30% words filtered by TF method

2 Term Clustering

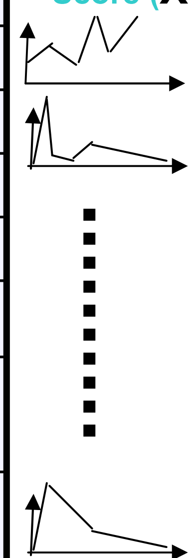
3 Co-occurrence Distribution Significance Score (X^2)

4 N-Gram collapsing

Frequent terms

All terms	tobacco	use	addict	…	X^2
tobacco	-	6	3	11	132
use	6	-		7	30
nictoine	8	5	5	2	342
expose	5	7	1	4	23
…	…	…	…	…	…
direct expose	2	5	1	7	258
smokeless tobacco	9	4	2	0	545

Co-occurrence matrix



Terms Clustering – Similarity Measures

Distribution-based Similarity

• Two terms are considered to be similar if they have similar co-occurrence distribution of co-occurrence with other terms.

• **Jensen-Shannon divergence value** of two terms indicates the distribution similarity.

$$J(w_1, w_2) = \log_2 2 + 1/\,2 \sum_{w^` \in G} \{h(P(w^`|w_1) + P(w^`|w_2)) - h(P(w^`|w_1)) - h(P(w^`|w_2))\}$$

Where

$h(x) = -x \log x, \quad P(w^`|w1) = freq(w^`,w1) / freq(w1)$

Pair-wise Similarity

• Two terms are assumed similar if they co-occur frequently.

• Pair-wise similarity is measured by **mutual information**

$$M(w_1, w_2) = \log \frac{P(w_1, w_2)}{P(w_1)\, P(w_2)}$$

Terms Clustering

Averaging hierarchical based clustering scores

1

Column terms

tobacco
use
…
addict

2

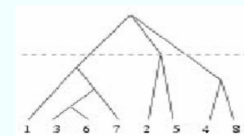
Similarity Measure Matrix

(Jensen Shannon divergence OR Mutual information)

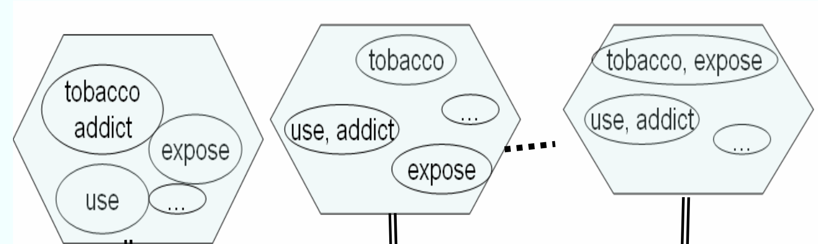
	tobacco	use	…	addict
tobacco	-	0.1	0.3	0.2
use	0.1	-	0.2	0.8
…	0.3	0.2	-	0
addict	0.3	0.8	0	-

3

Hierarchical Clustering



←Threshold (X%)



The choice of clustering Threshold dramatically affects the results.

Averaging over 10%-30% in 5% increments provides robust scores.

Co-occurrence distribution significance

4

Row terms

Clustered Column terms

	tobacco	use	addict	…
tobacco	-	6	3	11
use	6	-		7
nictoine	8	5	5	2
expose	5	7	1	4
…	…	…	…	…
direct expose	2	5	1	7
smokeless	9	4	2	0

Co-occurrence Distribution significance

	tobacco	use	addict	…
tobacco	-	6	3	11
use	6	-		7
nictoine	8	5	5	2
expose	5	7	1	4
…	…	…	…	…
direct expose	2	5	1	7
smokeless	9	4	2	0

Co-occurrence Distribution significance

$\sum$

5

Averaging scores over different clusters

6

Key words

Smokeless tobacco
nicotine
expose

Clique-based Terms Clustering

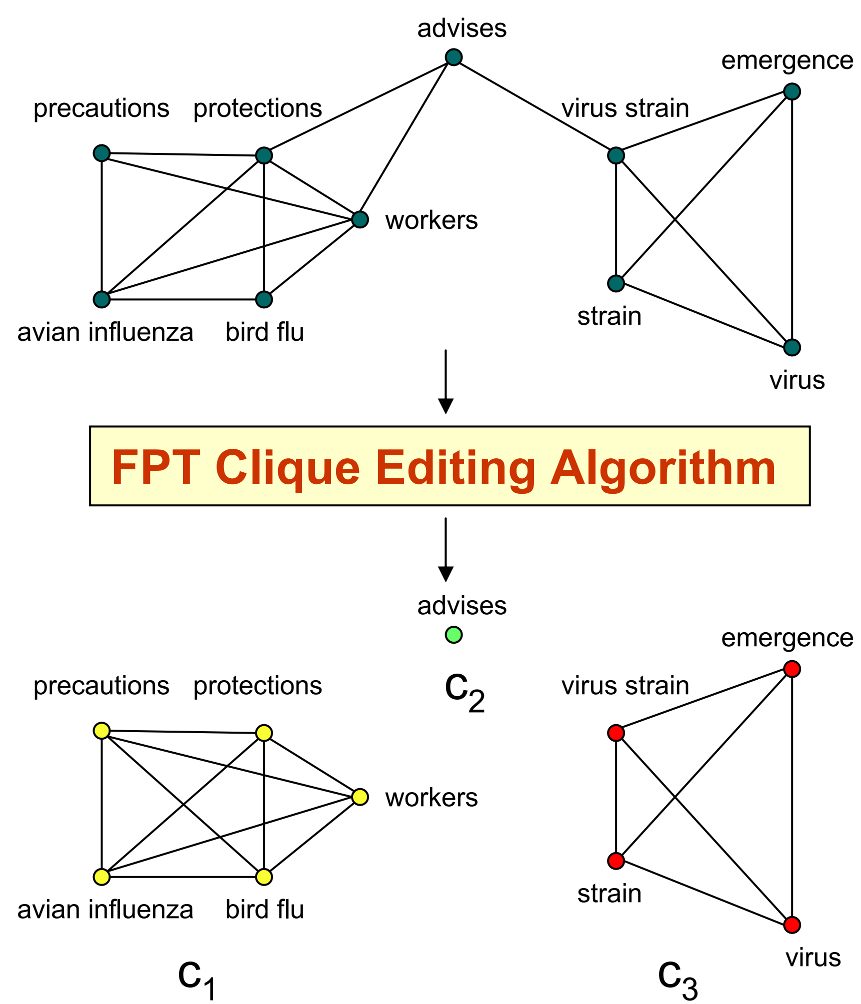
- The choice of clustering Threshold dramatically affects the results. Averaging partially solves this problem.

- Still, hierarchical clustering assigns each term to a single cluster – no overlaps. However, latent semantic meaning of terms should allow terms belong to multiple clusters.

- We developed a form of **clique-based** clustering based on our efficient **FPT clique editing algorithm**.

- **Benefits**:
  - No need to *a priori* specify the number of clusters (reducing the error due to Thresholding)
  - Overall quality of clusters is better or comparable with the averaging method
  - Comparable computational time on small/medium documents with the averaging method

Example



Salient Features –
Corpus Independent Algorithm

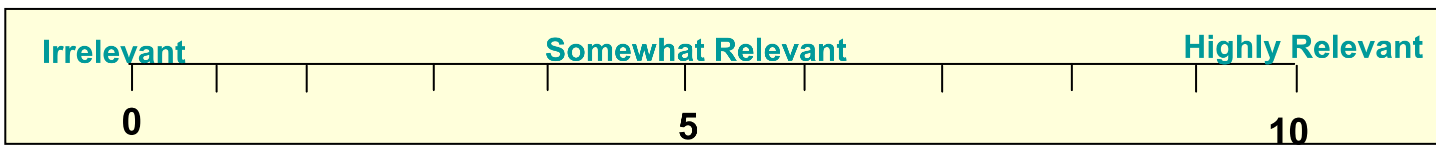
- With our corpus-independent algorithm, keyphrases overlap with those of the corpus-dependent approach at a rate of approximately 75% on documents targeted for inclusion into the BKC.
- More importantly, manual observation showed comparable results were obtained with both methods.
- No training documents required. An acceptable solution for bringing documents from a new domain. No need to “re-train” the system.

Evaluation of Key Phrases Extraction Methods

Document Collection:

Document Set	No of Documents
Aliweb	6
CSTR	12
Journal	6

Evaluation Method:



- **Top 15** key phrases extracted by each algorithm were selected for evaluation

- **Individual Key Phrase quality** – Each key phrase was scored according to its relevance to the document

- **Topic Coverage** – Entire key phrase set was evaluated for coverage of topic(s) in the document

Results – *Manual* Evaluation of Key Phrases

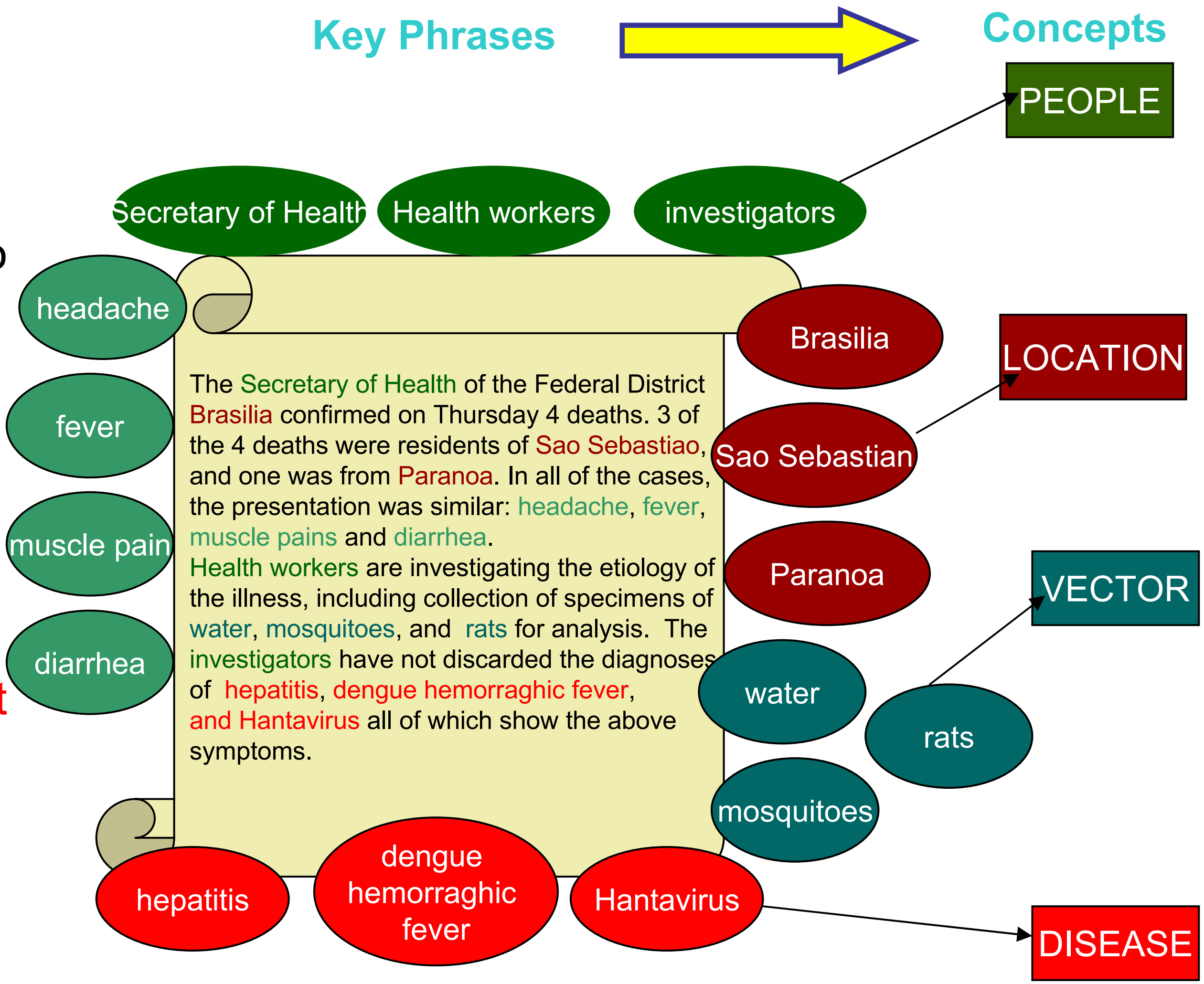
Based on independent evaluation by 6 users

Algorithm	Key Phrase Quality			Topic Coverage		
	Average	Std Dev.	Avg. Rank	Average	Std Dev.	Avg. Rank
Author Assigned	5.8	1.7	9	5.9	1.2	6.4
Corpus Dependent (with Domain Bayes)	4.9	1.2	8	6.6	0.6	8.4
Corpus Dependent (no Domain Bayes)	4.7	1.3	6.8	6.4	0.7	7.4
TF-IDF	4.6	1.3	5.9	5.9	1.2	6.4
TF	4.1	1.5	4.4	5.2	1.1	4.2
Corpus Independent	4.5	1.4	5.8	5.8	1.3	6.4

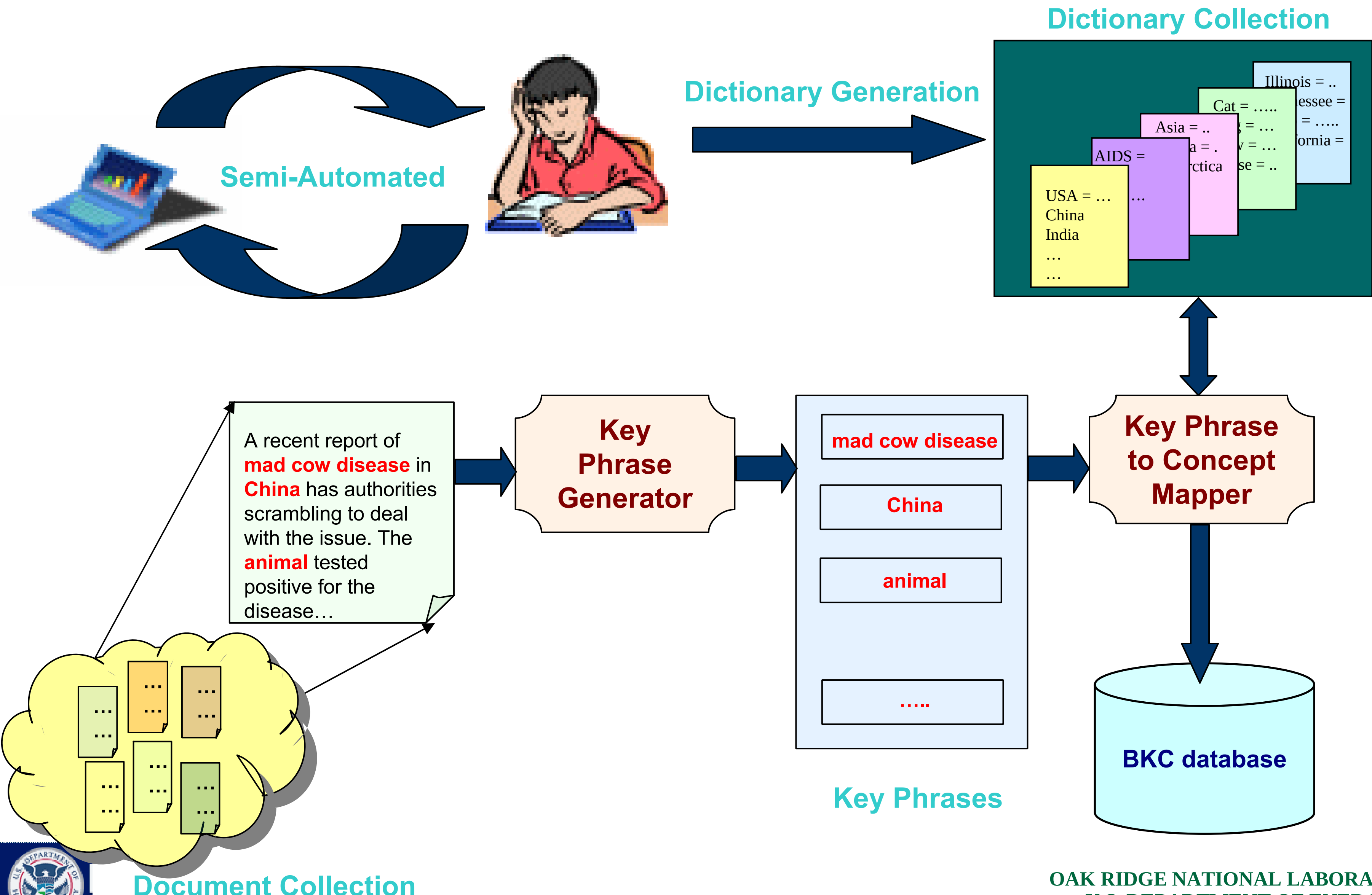
- Corpus Independent algorithm compares very well with Corpus Dependent ones. The results are very much identical to TF-IDF method.
- Corpus Independent algorithm could extract more human readable phrases than TF or TF-IDF method.
- Corpus Independent method outperforms TF method that is also a corpus independent method in all respects.

Concepts Mapping

- The greatest use of incorporating the power of key phrases is to identify their relevance to domains of interest.
- Challenges include:
  - Word sense disambiguation
  - Concept granularity
- Our approach targeted on building large concept dictionaries in a semi-automated way.

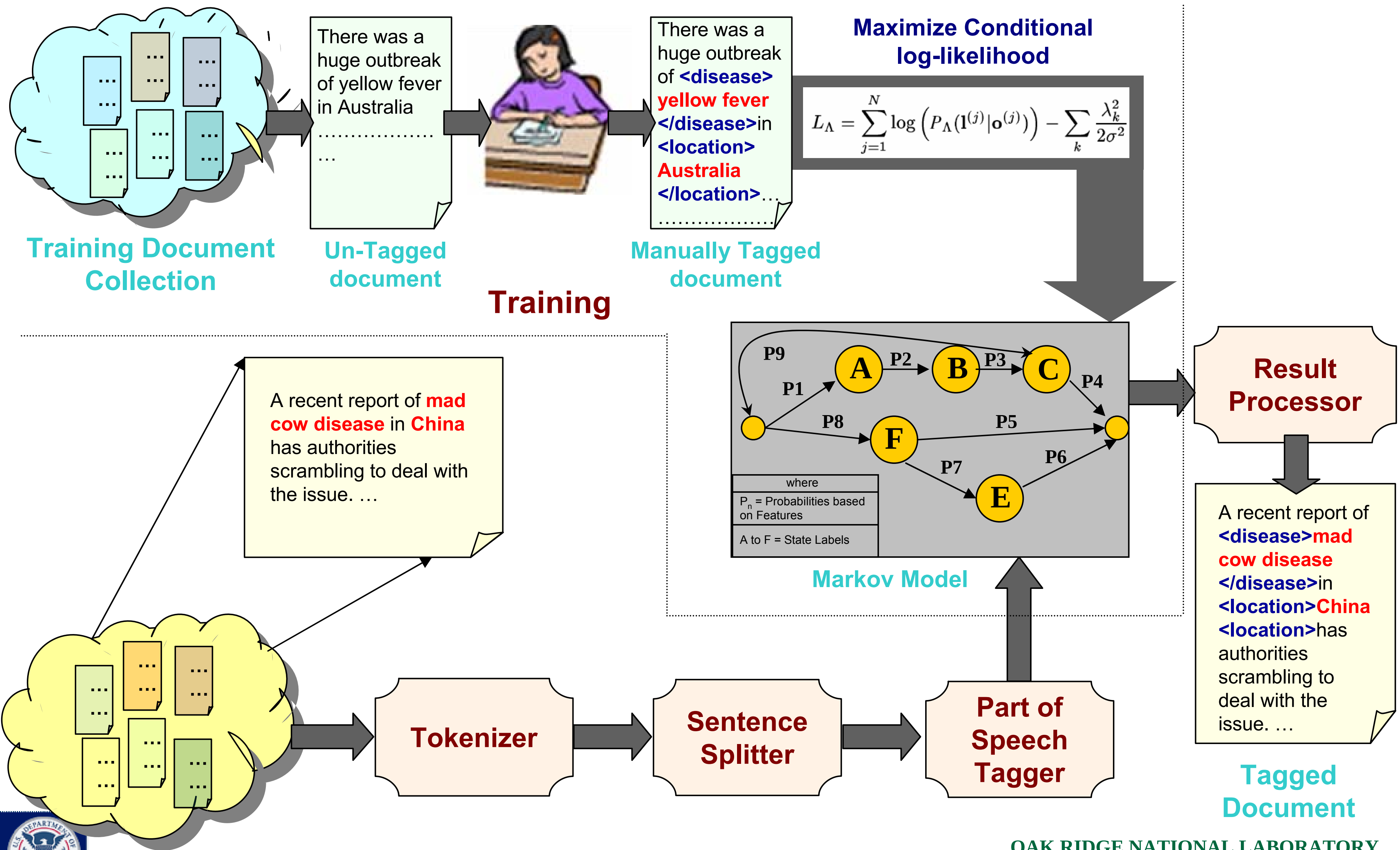


Mapping Key Phrases to Concepts in the BKC Semantic Graph



ORNL Named Entity Recognition Pipeline

Names, Dates, Locations, Diseases, … (in progress)



Software Infrastructure Delivered to BKC

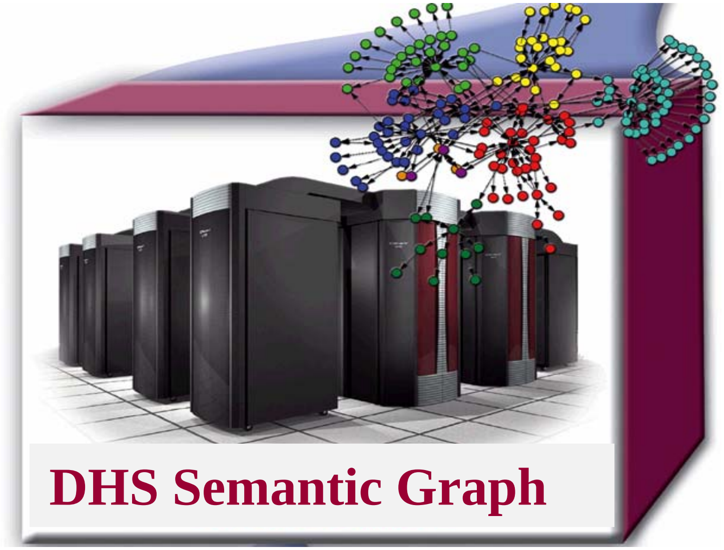
- Easy interface to keyword extraction package
  - Corpus Dependent Algorithm
  - Corpus Independent Algorithm (final package is due next week)
- Preprocessing tools packaged in *Java*
  - Sentence splitting
  - Stemmers – Lovins, Iterative *Lovins*
  - Anaphore Resolution
  - Part of speech tagging
  - Named entity recognition (in progress)
- Keyword extraction algorithm is implemented in *C*++ with following features
  - Dictionary based synonym collapsing and morphing package
  - Easy deployment of hierarchical term clustering tools using
    - Distribution similarity of terms
    - Pair-wise similarity of terms
- Shared CVS Repositories for easy code **sharing of ORNL source codes** to the LLNL team on the BKC project.

Intelligent Queries over Semantic Graphs

Processing of intelligent queries and advanced analysis of information in DHS presents a significant computational challenge.

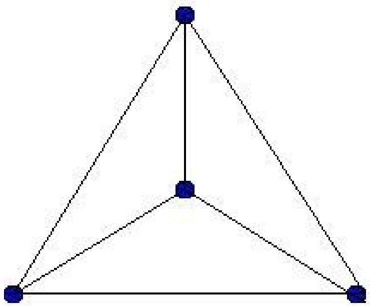
Example Queries beyond Google

- Identify a minimum group of people that are related to all the other people (**Minimum Vertex Cover**);
- Discover a suspicious pattern of interest in the DB (**Sub-graph Isomorphism**);
- Find the largest group of cities so that every two cities are affected by a disease spreading from one city to another or enumerate all such groups (**Maximum or Maximal Clique**);
- Extract the group of people and all relations between them that are common between two or more suspicious organizations (**Maximum Common Subgraph**).



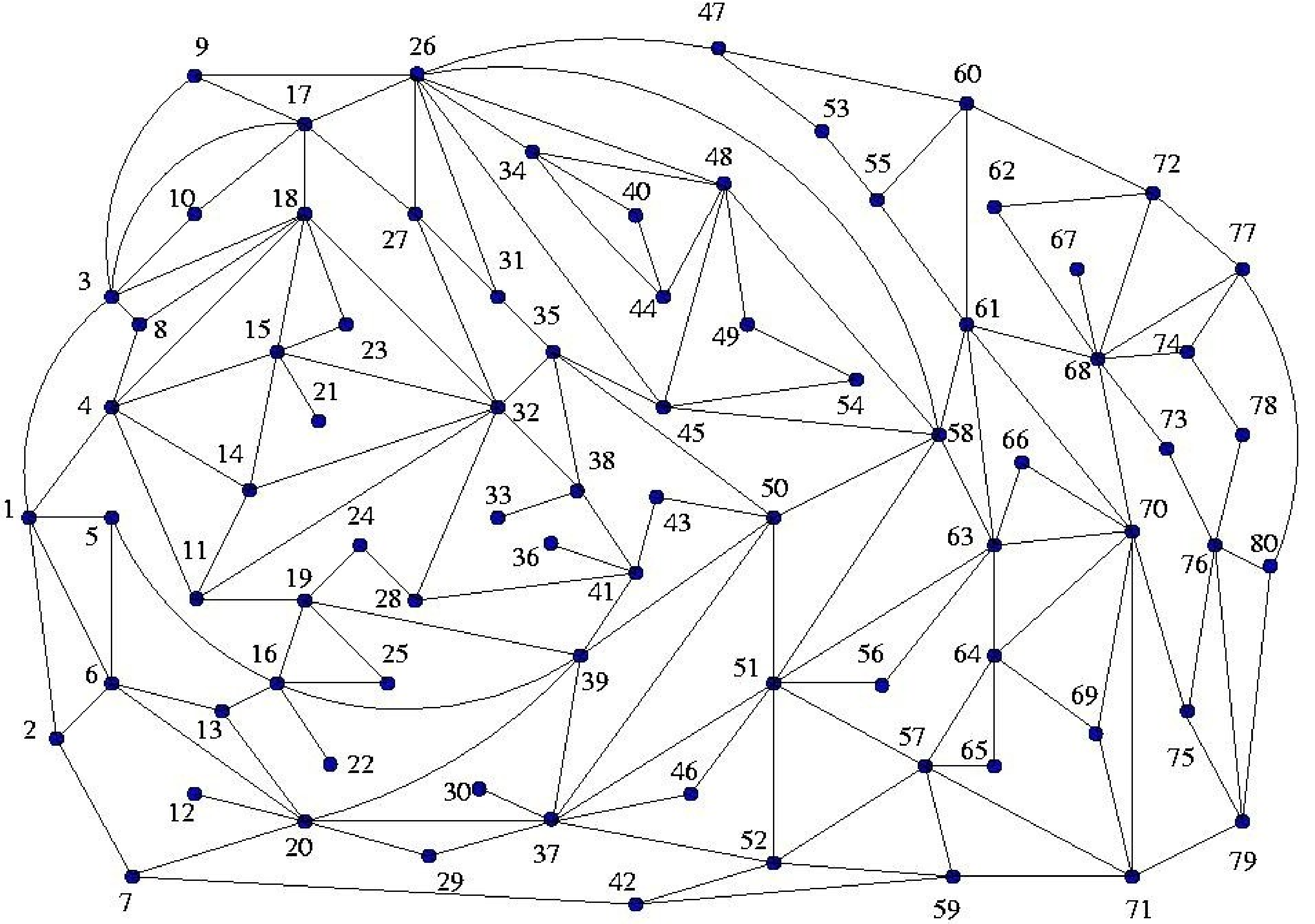
Example: Maximum Clique

- A clique is a complete subgraph, for example, K_4:

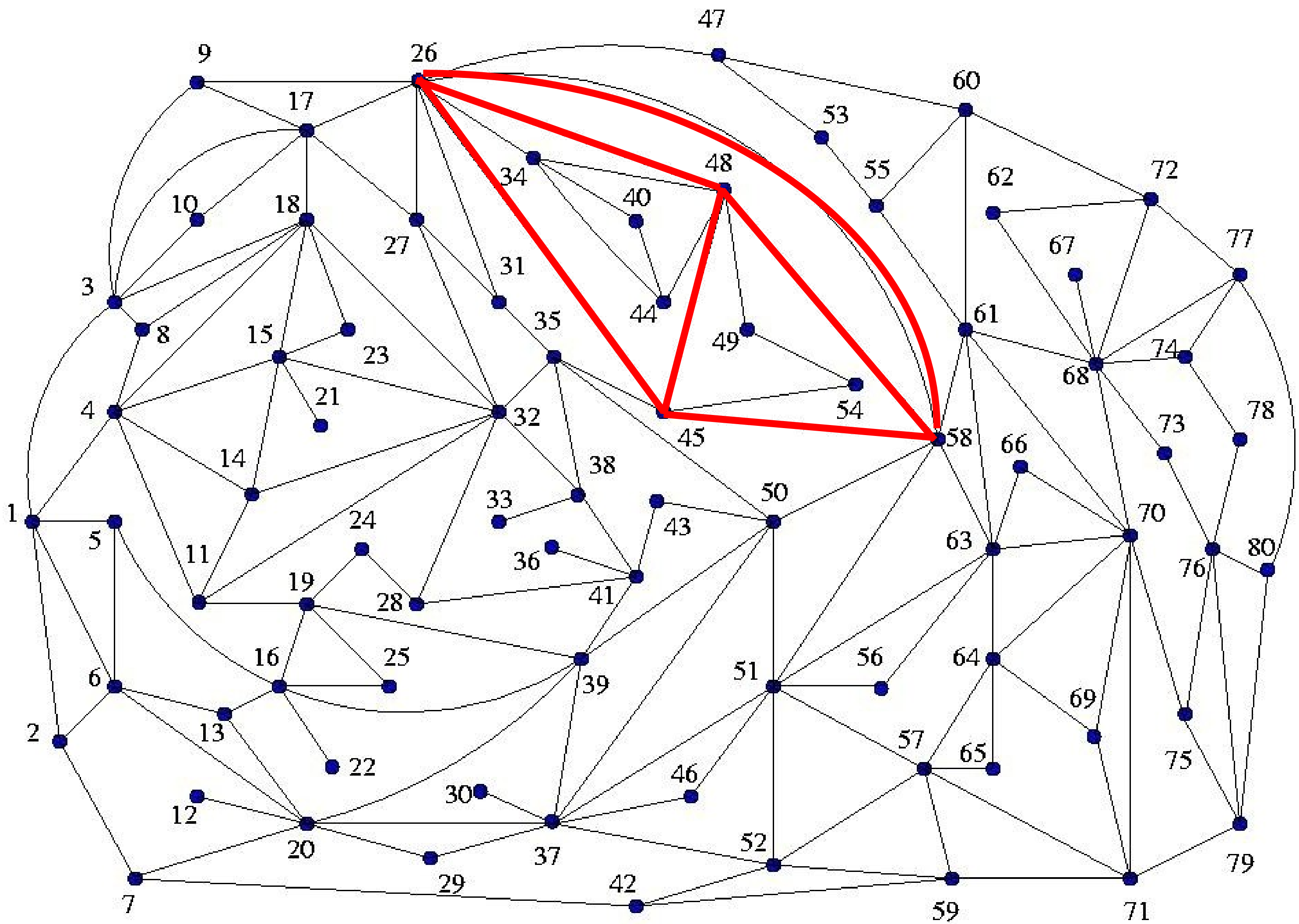


- Finding maximum clique in a graph is *NP*-complete problem, and difficult even for small cliques on planar graphs

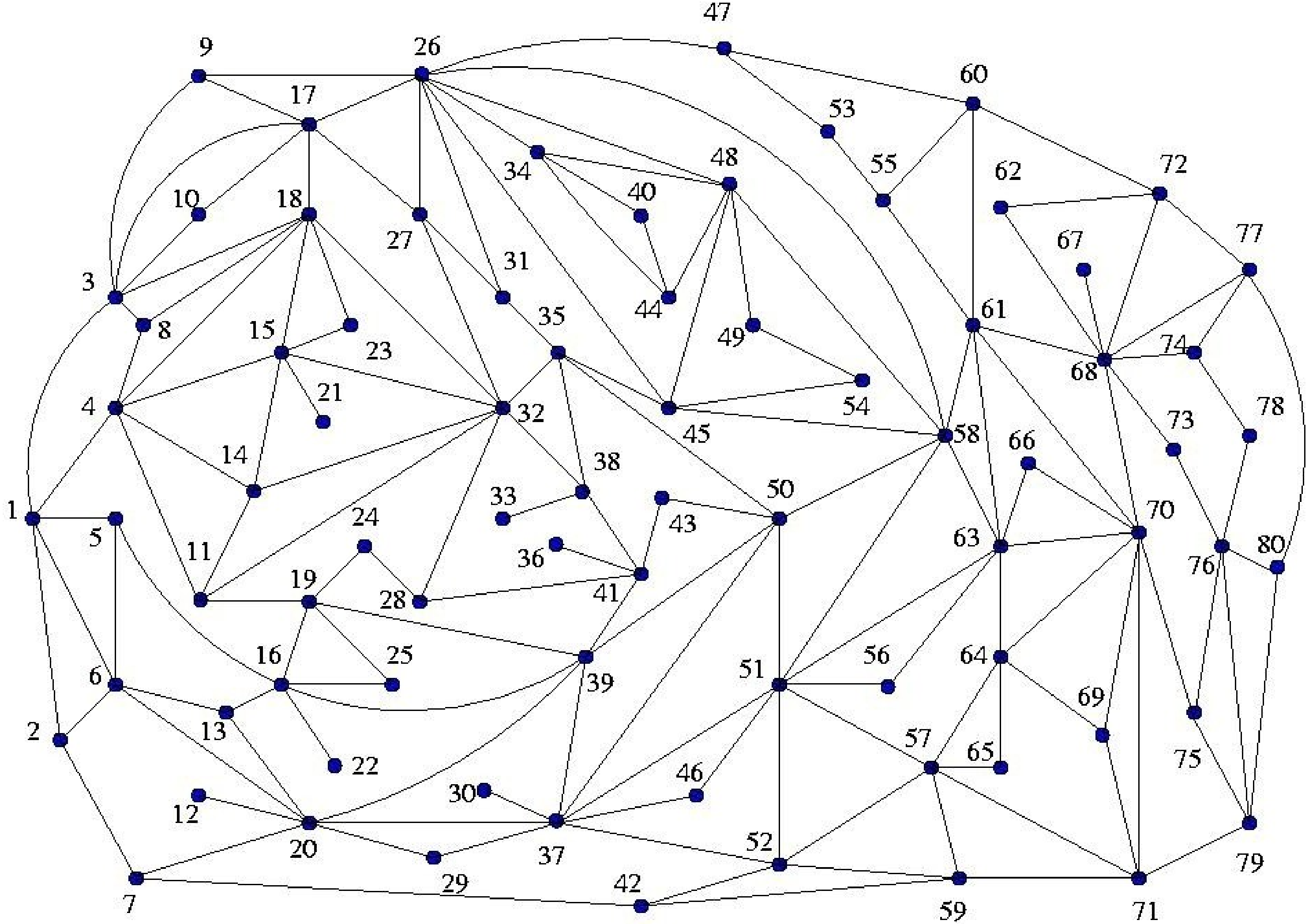
Does this graph contain K4?



Indeed it does!

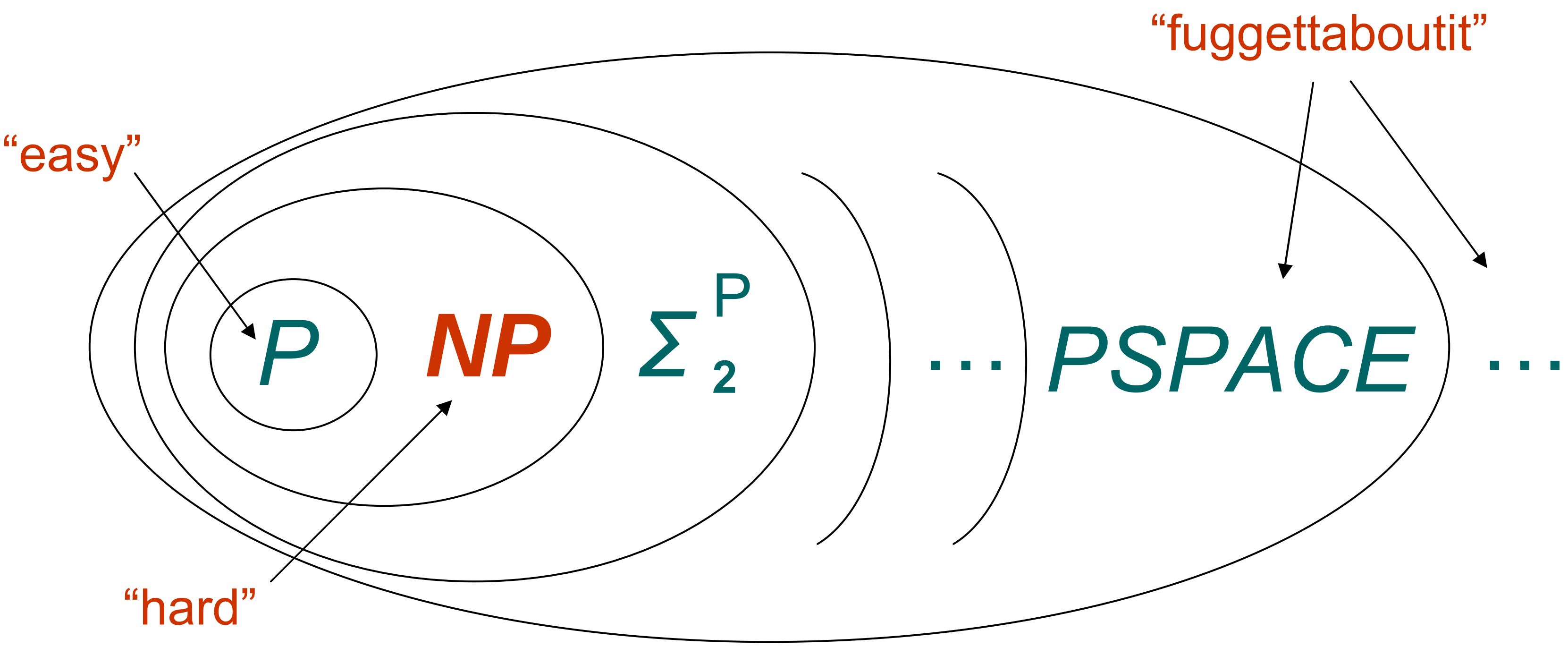


But, if it had not, what evidence would have been needed?



Classic Complexity Theory

- ***The Classic View:***

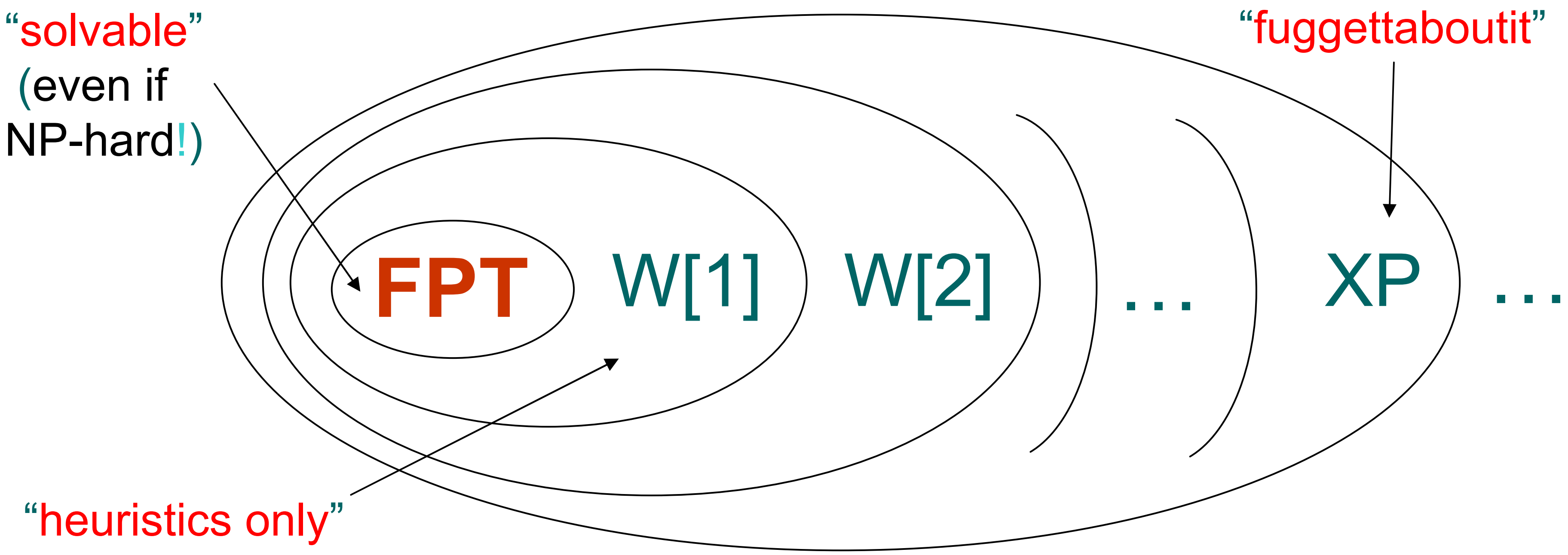


Parameter Sensitivity: Instance(n,k)

- **Suppose our problem is, say, *NP*-complete.**
- **Consider an algorithm with a time bound such as $O(2^{k+n})$.**
- **And now one with a time bound more like $O(2^k + n)$.**
- **Both are exponential in parameter value(s).**
- **But what happens when *k* is fixed?**

Parameterized Complexity Theory

Hence, the Parameterized View:



Fixed Parameter Tractability

- Fixed Parameter Tractability offers extremely efficient methods of **reducing the search space** for a certain subclass of *NP*-complete problems, known as FPT.

- FPT branching techniques also offer an **effective method of parallelizing** difficult problems:
  - Embarrassingly parallel
  - Little or no communication between processors

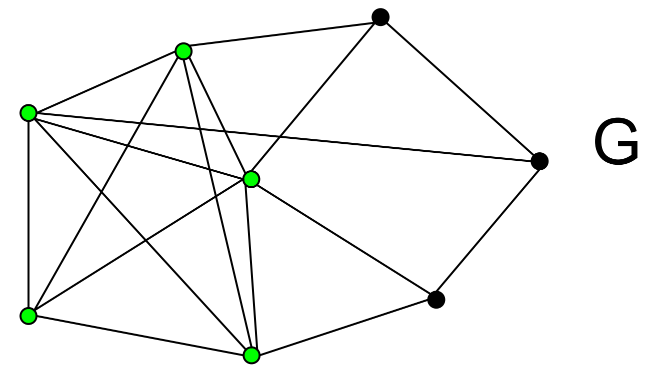
- These techniques have lead to the implementation of the **world's fastest codes** for solving these two well-known NP-complete problems.

Clique → Vertex Cover

Reduction:

- The Maximum Clique is **not** FPT
- Fortunately, Vertex Cover **is** FPT
- Vertex Cover is a **complementary dual** to Clique

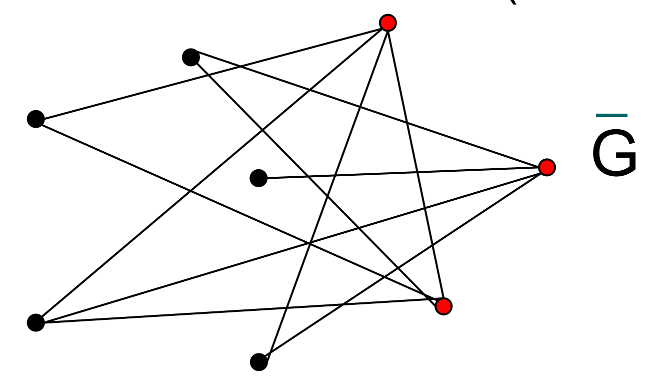
Maximum Clique (Size 5)



Vertex Cover - Major Steps:

- **preprocess** via degree structures
- **kernelize** to computational core
- parallel **branching** explores core
- **interleave** all three

Minimum Vertex Cover (Size 3)



Performance Results

Graph Name	Graph Size	Cover Size	Instance Type	Sequential Kernelization	Sequential Branching	Parallel Branching	Dynamic Decomposition
Set-1	839	399	Yes	34 seconds	7 seconds	Not needed	Not needed
Set-2	839	398	No	34 seconds	141 minutes	82 minutes	20 minutes
Set-3	2466	2044	Yes	203 minutes	~ 5 days	~ 5 days	140 minutes
Set-4	2466	2043	No	203 minutes	6+ days	6+ days	620 minutes

So clique size is 422. **A direct assault ~ 2466^{422}.**

32 PEs @ 500MHz.
Load balancing is critical.
"No" is harder than "yes."

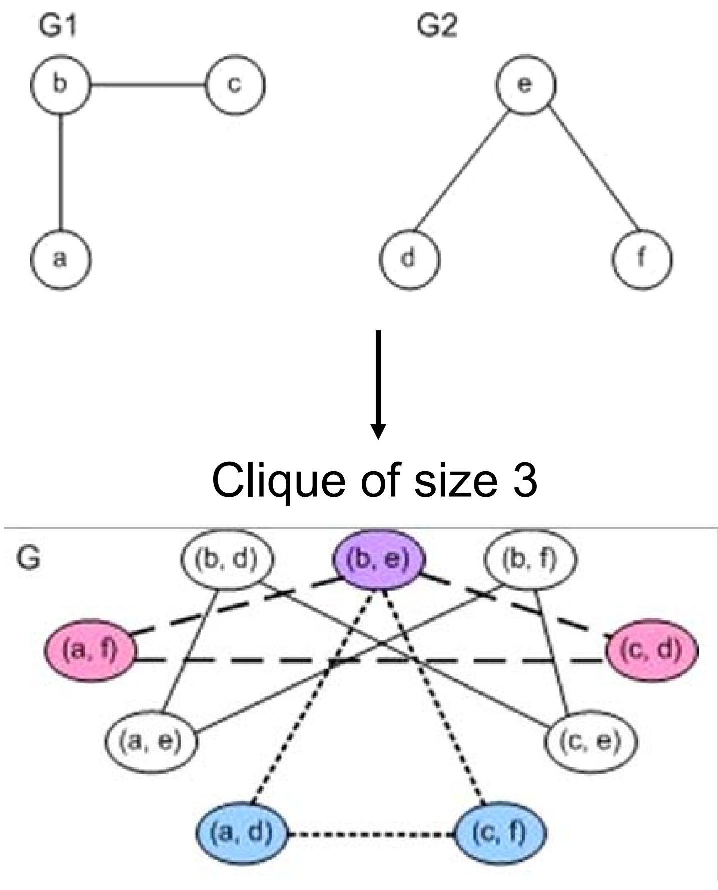
Results on Big Graphs

- 12,422 (vertices)
- Over 100M edges
- ~6 several days of parallel CPU time
- But a direct assault would have been ~12422^{369}.

Graph Matching → Clique

- Maximum Common Subgraph (MCS) and Subgraph Isomorphism are special cases of Graph Matching.
- Existing approaches to MCS:
  - Clique-based (Bron-Kerbosch, Robson); ***$O(1.19^{mn})$***
  - Backtracking (McGregor, Krissinel); ***$O(m^{n+1}n)$***
  - Dynamic programming (Akutsu) (trees of bounded degree)
- MCS is **not** FPT. But we solve MCS by reducing it to Clique on the ***association graph***.
- Our method is the fastest known on general graphs with ***$O((m+1)^n)$*** but much better in practice since there are much less choices for branching than $(m+1)$

MCS of size 3



Clique of size 3

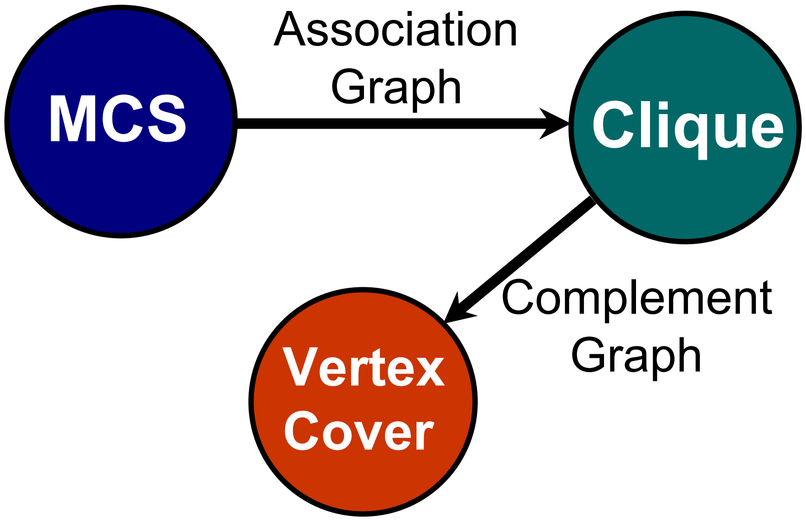
Association Graph

ORNL Scalable Algorithms for Semantic Graphs

Prototyped the library of scalable parallel graph matching algorithms for NP-hard graph problems with polynomial time solution.

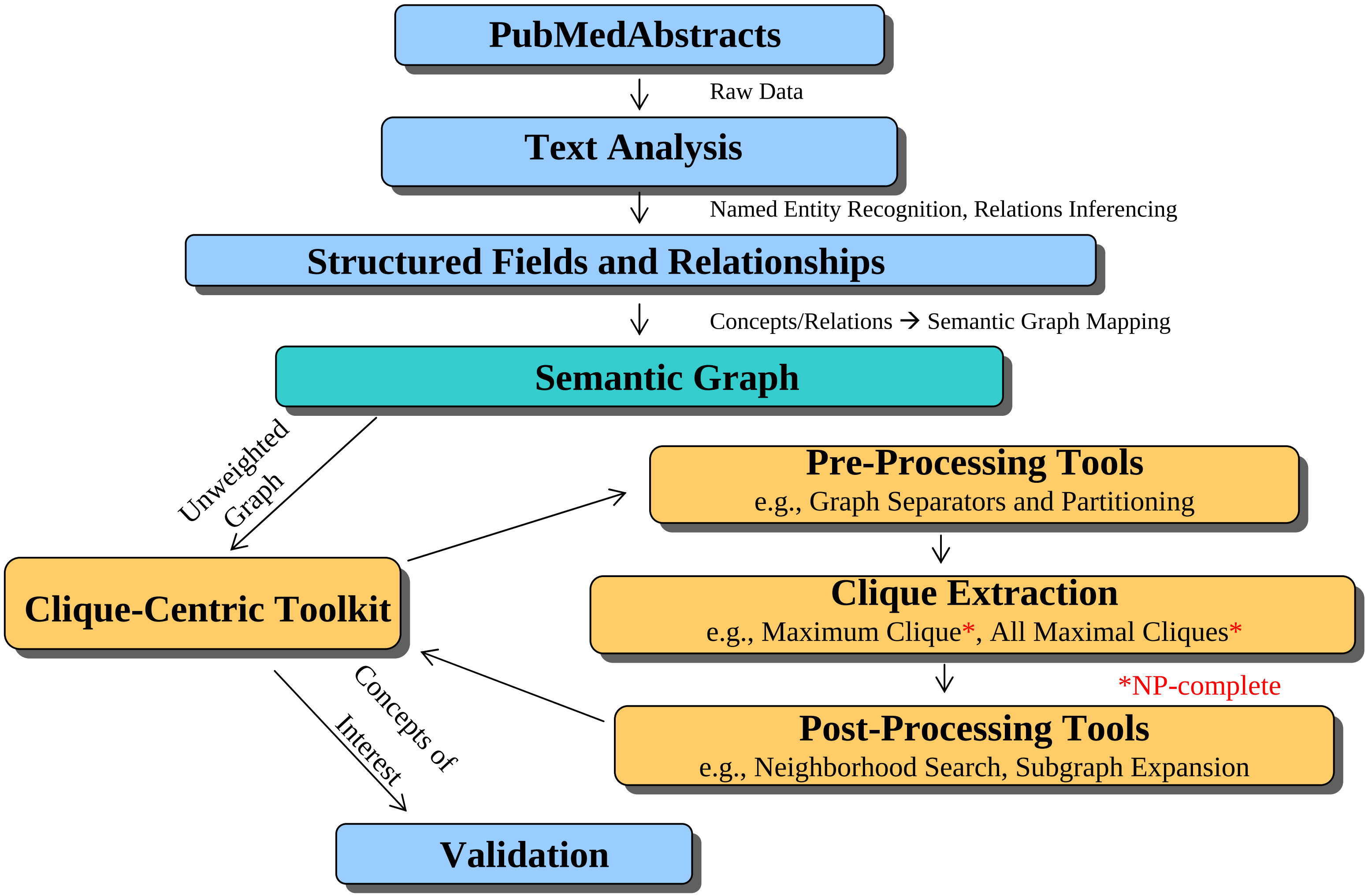
Library Features:

- **Exact polynomial** solutions via **Fixed Parameter Tractability** (FPT) reduction:
  - Minimum Vertex Cover (VC)
  - Sub-graph Isomorphism (SI)
  - Maximum or Maximal Clique (Clique)
  - Maximum Common Subgraph (MCS)
- The **fastest and most scalable** (in problem size) than reported in literature.
- Supports different types of graphs: directed, undirected, labeled, and unlabeled.



<u>Example Semantic Graph</u>:
12,422 vertices and >100M edges
Maximum Clique: 399 vertices

Putting it altogether…



Summary of FY-04 Accomplishments

- Developed novel algorithms for key phrases extraction, weighting, and concepts mapping. Integrated them into the BKC pipeline.
- Analyzed and extracted information from the following BKC-related free-text sources:
  - ProMED Mail (21,000 e-mails)
  - PubMed (105,978 abstracts)
  - IAIP (10683 reports categorized by sectors)
- The text analysis pipeline included:
  - Corpus-dependent and corpus-independent key phrases extraction
  - Mapping extracted key phrases into concepts of BKC semantic graph
  - Daily ingest and specialized parsing of target data sources
  - XML representation and upload of structured text into the BKC database
- Prototyped the library of parallel and scalable graph algorithms:
  - Maximum and Maximal Cliques
  - Minimum Vertex Cover
  - Maximum Common Subgraph
  - Subgraph Isomorphism

Summary

<u>Goal</u>: Provide a capability for automated mapping of unstructured free text to Semantic Graph and for efficient query over Semantic Graph.

- **Motivation**
  - The construction of the concept graphs from unstructured text is a very labor intensive and tedious task that requires automation.
  - Semantic graph queries are often NP-complete
- **Major accomplishments**
  - Intelligent text preprocessing
  - Advanced methods for concepts extraction, scoring, and mapping
  - Scalable graph algorithms over semantic graphs
- **Benefits**
  - Facilitate free text data feed to the Texas semantic graph.
  - Discover advanced knowledge from the semantic graph.